

MODUL 4

MEMBACA DAN MENULISKAN DATA KE FILE ASCII

I. Tujuan Praktikum

Tujuan praktikum adalah sebagai berikut:

- Mahasiswa mampu membaca dan menuliskan matriks 2D dari/ke file ASCII.
- Mahasiswa mampu membandingkan dan memproses dua file ASCII dengan Python.

II. Teori Dasar

File umumnya dikenal dan dibedakan menjadi dua yaitu file ASCII dan biner. ASCII atau *American standard code for information interchange* adalah standar internasional dalam kode huruf dan simbol seperti hex karena asal mulanya dibangun dari kode telegram namun lebih bersifat universal, contohnya dalam desimal 122 untuk z. Jumlah kode ASCII dalam desimal ada 255 kode, 0 s.d 127 menunjukkan teks sedangkan 128 s.d 255 merupakan kode untuk memanipulasi grafik. ASCII selalu digunakan komputer untuk menunjukkan teks. Karena komputer hanya mengenal angka (biner) maka ASCII adalah representasi numerik dari sebuah karakter. Kode ASCII memiliki komposisi bilangan biner sebanyak 7 bit.

File ASCII sendiri adalah file teks yang setiap bitnya merepresentasikan satu karakter yang tercatat pada kode ASCII. File ASCII tidak memiliki format bold, italic, tab, dan seterusnya. Biasanya file ASCII berekstensi file berupa txt. File ASCII bisa disebut sebagai file biner yang disimpan dalam kode ASCII, ada 128 kode ASCII (masing-masingnya besarnya 7 bit) yang khusus digunakan untuk menunjukkan karakter ASCII.

File selain berfungsi sebagai input atau output suatu program juga biasanya dimanfaatkan untuk menyimpan data. Data disimpan kedalam file agar tidak membebani memori yang sedang bekerja, sehingga dapat mempercepat waktu komputasi.

Tabel 4.1. Tabel hubungan karakter ASCII dengan desimal dan heksadesimal

Karakter	Nilai Unicode (heksadesimal)	Nilai ANSI ASCII (desimal)
NUL	0000	0
SOH	0001	1
STX	0002	2
ETX	0003	3
EOT	0004	4
ENQ	0005	5

Ketika sedang mengetik di file text menggunakan text editor (Notepad) maka proses itu disebut sebagai pengeditan file ASCII. Misalnya diketik karakter “c” pada notepad, maka itu dibaca sebagai kode ASCII heksadesimal sebagai 63, atau desimal sebagai 99, kemudian di konversi ke biner menjadi 0110 0011 (7 bit), disebut hanya 7 bit karena paling depan tidak digunakan untuk merepresentasikan kode ASCII (nilainya 0).

Tabel 4.2. Tabel hubungan karakter ASCII dengan heksadesimal dan binary

ASCII	'c'	'a'	't'
Hex	63	61	74
Binary	0110 0011	0110 0001	0111 1000

File biner adalah file komputer yang bukan sebuah file teks, filenya bisa langsung dimengerti oleh komputer. File-file objek, suara, gambar dan banyak format merupakan contoh file biner. File biner dimanfaatkan karena setiap bit dari file biner dapat menjadi 1 dari 256 susunan bit, jadi tidak terbatas seperti kode ASCII.

File-file ASCII dan biner bisa dipanggil dengan menggunakan Python untuk kemudian diolah lebih lanjut. Sementara file biner sendiri bisa langsung dimengerti oleh komputer. Beberapa hal yang akan dipelajari pada modul ini adalah

1. Membaca isi file, baik itu string maupun angka dari file ASCII dan menampilkannya ke konsol.
2. Mengolah data yang telah dibaca dari file ASCII.
3. Membandingkan dua buah file ASCII.
4. Menyimpan data hasil olahan baik dalam bentuk string maupun angka di file ASCII.

Pada Python ada dua pembacaan, pembacaan tingkat tinggi dan tingkat rendah. Pembacaan tingkat biasanya menggunakan package Python berupa **numpy.genfromtxt** dan **numpy.savetxt** sedangkan untuk pembacaan tingkat rendah menggunakan fungsi **var = open("nama_file.txt","mode_akses")**, kemudian juga **var.close()**, **var.write()**, dan **var.read()**. Sementara mode akses berisi **r** hanya untuk membuka file, **rb** untuk membuka file biner, **r+** untuk membaca dan menulis, **w** hanya untuk menulis, **w+** untuk menulis dan membaca serta menimpa file yang ada jika file tersebut ada, jika file tidak ada, buat file baru untuk membaca dan menulis, dan lain-lain. Sedangkan **var** sendiri merupakan variable bebas yang bisa diganti dengan huruf apapun.

III. Tugas Pendahuluan

1. Apa perbedaan file ASCII dengan file biner, selain yang disebutkan di modul?
2. Tuliskan kode ASCII dalam heksadesimal dan desimal dari karakter ASCII #, 9, @, H, h, dan IND!
3. Ubahlah karakter ASCII didalam petik "MetK0m" menjadi biner 8 bit!
4. Jelaskan perbedaan fungsi antara **numpy.genfromtxt**, **numpy.savetxt**, **var=open()**, **var.write()**, **var.read()**, dan **var.close()**!
5. Buatlah flowchart membaca, mengolah dan menuliskan suatu data ke file ASCII!

IV. Langkah Praktikum

1. Membaca, mengolah, dan menyimpan file dengan menggunakan pembacaan tingkat rendah. Simpan dengan nama kode **pm4a.py** berikut

```
fo = open("file4a.txt", "w")
fo.write("Python is a great language.\nYeah its great!!\n")
fo.close()

fo = open("file4a.txt", "r+")
str = fo.read(10)
print("Read string is : ", str)
fo.write("Write string again!!\n")
fo.close()
```

2. Buatlah file ASCII secara manual di notepad dengan diketik terlebih dahulu dan kemudian diberi nama **prak4a.txt** yang berisi:

1	1	2	3
2	4	5	6
3	7	8	9

Kemudian **prak4b.txt** yang berisi:

1	1	1	1
2	2	2	2
3	3	3	3

3. Kemudian baca, olah, dan simpan file dengan menggunakan pembacaan tingkat tinggi dari kedua data tersebut dengan menggunakan kode **pm4b.py** berikut

```
import numpy as np

X = np.genfromtxt('prak4a.txt')
Y = np.genfromtxt('prak4b.txt')
print(type(X))
result = [[0, 0, 0],
           [0, 0, 0],
           [0, 0, 0]]

for i in range(len(X)):
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]

for row in X:
    for elem in row:
        print("%.2f" % elem, end=' ')
    print()
print(20 * '=')

np.savetxt('prak4oa.txt', result, fmt='% .0d % .2f % .3f')
```

4. Buatlah file ASCII dengan nama **prak4c.txt** yang berisi

1	Log	Lat	Elev	Rawgrav
2	105.2626	-5.453	3.704	1567.29
3	105.2475	-5.4459	26.123	1552.658
4	105.2488	-5.4636	6.987	1566.629
5	105.2364	-5.4631	24.549	1558.902
6	105.2344	-5.4902	9.071	1577.052
7	105.2626	-5.453	3.704	1595.423

Contoh data diatas adalah data gravity dengan Rawgrav berupa data bacaan alat gravity (belum mGals). Kemudian lakukan perhitungan dibawah ini, hasilnya tampilkan di file **prak4ob.txt**.

Percepatan gravitasi (PG) = ((Rawgrav-1500) + 1530,844) x 1.00043726.

Koreksi Udara Bebas (KUB) = 0.03068 x Elev.

Rata-rata Elevasi (RE) = Jumlah semua elev/banyaknya elevasi.

SDRE = Elev – RE.

Dengan menggunakan kode **pm4c.py** berikut

```
import numpy as np
# Read data
data = np.genfromtxt('prak4c.txt')
Ld = len(data)
Log = data[1:Ld, 0]
Lat = data[1:Ld, 1]
Elev = data[1:Ld, 2]
Rawgrav = data[1:Ld, 3]
# Operate data
PG = ((Rawgrav - 1500) + 1530.844) * 1.00043726
KUB = 0.03068 * Elev
RE = sum(Elev) / len(Elev)
SDRE = Elev - RE
# Save data
LL = len(Log)
data_final = [Log, Lat, Elev, Rawgrav, PG, KUB, SDRE]
data_final = np.transpose(data_final)
print(data_final)
np.savetxt('prak4ob.txt', data_final, fmt='%.2f')
```

5. Membandingkan dua buah file berikut:

Gunakan kode berikut ini untuk membangkitkan dua file teks yang berisi bilangan random, beri nama file tersebut sebagai **orand1.txt** dan **orand2.txt**. Salin kode **pm4d.py** berikut.

```
import numpy as np
from numpy import random as rd

Z1 = []
for i in range(10):
    Z1.append(rd.randint(1, 10))
np.savetxt('orand1.txt', Z1, fmt='%d')

Z2 = []
for j in range(10):
    Z2.append(rd.randint(1, 10))
np.savetxt('orand2.txt', Z2, fmt='%d')
```

Kemudian cari data yang sama antara keduanya dan tampilkan di konsol dan di file **outfileCompare.txt**. Salin kode dibawah dan simpan sebagai **pm4e.py**.

```
import numpy as np

orand1 = np.genfromtxt('orand1.txt')
orand2 = np.genfromtxt('orand2.txt')
result = []
m = 0
print('No\tOrand1\tNo\tOrand2')
for i in range(len(orand1)):
    for j in range(len(orand2)):
        cek = int(orand1[i, 1]) - int(orand2[j, 1])
        if cek == 0:
            result.append([orand1[i, 0], orand1[i, 1],
                           orand2[j, 0], orand2[j, 1]])
            print(orand1[i, 0], '\t', orand1[i, 1], '\t',
                  orand2[j, 0], '\t', orand2[j, 1])
            m = m + 1
np.savetxt('ofileCompare.txt', result, fmt='%d')
print('Data yang sama sebanyak: ',m)
```

V. Laporan dan Analisis

Tugas yang harus dilampirkan pada laporan berupa:

1. Lakukan prosesi gravity memanfaatkan Python dengan cara load **datamentah.txt** dan menggunakan persamaan yang telah diberikan. Kemudian simpan hasilnya berupa Northing, Easting, Elev, mGal, TDCf, FAC, BC dan CBA dengan nama file **procesgravity.txt**.

Misal	Keterangan	Tampilan Keluaran di file text (procesgravity.txt)
TDC = 0	Tidal Correction (TDC)	Northing Easting Elev mGal TDCf FAC BC CBA
TDCf = mGal – TDC	Tidal Corected (TDCf)	
DCf = TDCf– DC	Bacaan alat miliGal (mGal)	
DG (n) = DCf(n) – DCf(1)	Drift Corrected (DCf)	
Const = 977167.257528	Delta Gobs (DG)	
AG = Cons + DG	Drift Correction (DC)	
FAC = 0.3086 x Elev	Abs Gobs (AG)	
rho = 2.5	Free Air Correction (FAC)	
BC = 0.04193*rho*Elev	Elevasi (Elev)	
TC = 0	Bougeur Correction (BC)	
CBA = AG-(Gn-FAC+BC-TC)	Densitas batuan rata2 (rho)	
	Tirrain Corection (TC)	
	Complete Bougeur Anomaly (CBA)	

2. Bandingkan data-data yang memiliki titik pengukuran yang sama pada file **databanding1.txt** dan **databanding2.txt**. Kemudian tampilkan titik-titik yang sama pada file txt dengan nama **outbanding.txt**.

~ Selamat Praktikum ~